

(19)日本国特許庁(JP)

(12) 公開特許公報(A)

(11)特許出願公開番号

特開平6-266596

(43)公開日 平成6年(1994)9月22日

(51)Int.Cl.<sup>5</sup>

G 0 6 F 12/00

識別記号

5 2 0 P 8944-5B

5 5 0 9366-5B

庁内整理番号

F I

技術表示箇所

審査請求 未請求 請求項の数11 O L (全 21 頁)

(21)出願番号 特願平5-51041

(22)出願日 平成5年(1993)3月11日

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72)発明者 片山 国弘

神奈川県横浜市戸塚区吉田町292番地 株式会社日立製作所マイクロエレクトロニクス機器開発研究所内

(72)発明者 柿 健一

神奈川県横浜市戸塚区吉田町292番地 株式会社日立製作所マイクロエレクトロニクス機器開発研究所内

(74)代理人 弁理士 富田 和子

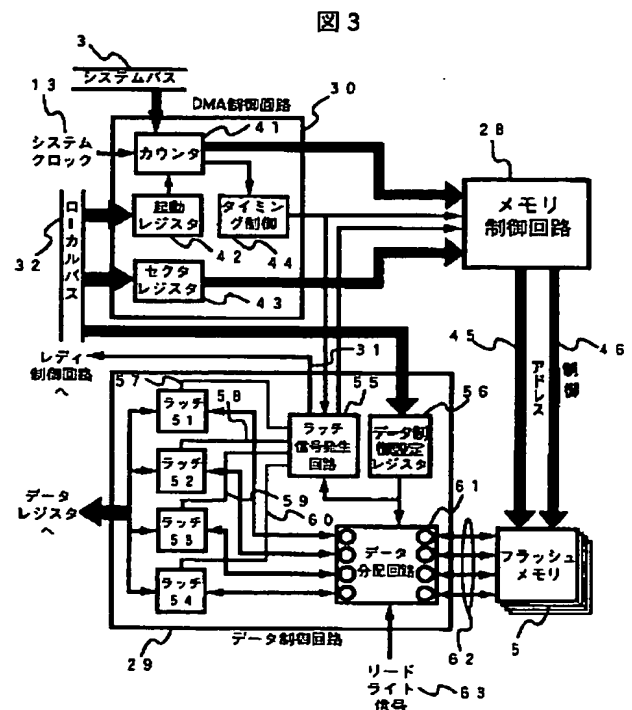
最終頁に続く

(54)【発明の名称】 フラッシュメモリファイル記憶装置および情報処理装置

(57)【要約】 (修正有)

【構成】 Xビットのデータバスを介し最小消去単位がXビットより大きくデータアクセス幅がxビット ( $x = X/p$ : pは2以上の整数)であるフラッシュメモリ素子を用いるファイル記憶装置において、それぞれ同時にアクセス可能なp組のフラッシュメモリ素子群により構成したフラッシュメモリ装置5、データバス上のデータを少なくともp分割する分割手段51~54、少なくとも、p個のxビットデータをフラッシュメモリ素子群の1組に対応付ける第1の機能、及びp個のxビットデータをそれぞれフラッシュメモリ素子群の別個の組に対応付ける第2の機能とを有するデータ分配手段61、並びにアクセス対象ファイルのファイル管理の記憶容量単位の数に応じて、第1及び第2の機能を切替るようデータ分配手段61を制御する制御手段56を備える。

【効果】 記憶エリアを有効に利用しながら複数バイト並列同時書き込みが可能になる。



BEST AVAILABLE COPY

## 【特許請求の範囲】

【請求項1】ファイルのデータをXビットのデータバスを介して格納するファイル記憶装置であって、最小消去単位が前記Xビットより大きくデータアクセス幅がxビット ( $x = X/p$  : pは2以上の整数) であるフラッシュメモリ素子を用いるフラッシュメモリファイル記憶装置において、

それぞれ同時にアクセス可能なp組のフラッシュメモリ素子群により構成したフラッシュメモリ装置と、

前記Xビットのデータバス上のデータを少なくともp分割する分割手段と、

少なくとも、該分割により得られたp個のxビットデータを前記フラッシュメモリ素子群の1組に対応付ける第1の機能と、前記p個のxビットデータをそれぞれ前記フラッシュメモリ素子群の別個の組に対応付ける第2の機能とを有するデータ分配手段と、

アクセス対象ファイルのファイル管理の記憶容量単位の数に応じて、前記第1および第2の機能を切替るよう前記データ分配手段を制御する制御手段と、

を備えることを特徴とするフラッシュメモリファイル記憶装置。

【請求項2】前記制御手段は、前記フラッシュメモリ素子の最小消去単位のメモリエリア内には異なるファイルのデータが混在しないよう、前記データ分配手段を制御することを特徴とする請求項1記載のフラッシュメモリファイル記憶装置。

【請求項3】前記制御手段は、前記ファイルの連続するp個の記憶容量単位のデータに対して前記第2の機能を選択し、1個の記憶容量単位のデータに対して前記第1の機能を選択するよう、前記データ分配手段を制御することを特徴とする請求項1または2記載のフラッシュメモリファイル記憶装置。

【請求項4】前記記憶容量単位と、前記フラッシュメモリ素子の最小消去単位が等しいことを特徴とする請求項1または2記載のフラッシュメモリファイル記憶装置。

【請求項5】プログラムおよびデータを処理する中央演算処理手段と、該中央演算処理手段を駆動するクロック発振手段と、フラッシュメモリを記憶媒体としたファイル記憶手段と、該ファイル記憶手段のフラッシュメモリのアクセスを制御するファイル記憶制御手段を構成要素とする情報処理装置において、

前記クロック発振手段が発生するクロック信号と同一の信号あるいは同期する信号を前記ファイル記憶制御手段に入力し、前記中央演算処理手段と該ファイル記憶制御手段が同期動作を行ってファイルデータを送受することを特徴としたフラッシュメモリファイル記憶装置を搭載する情報処理装置。

【請求項6】前記中央演算処理手段と前記ファイル記憶手段の1回のデータアクセス幅が異なる場合、前記ファイル記憶制御手段はファイルデータの処理中か処理完了

かを示す状態提示信号を出力する手段を備え、前記中央演算処理手段は前記状態提示信号を受け、これが処理中であることを示していれば処理終了までの期間、処理の進行を停止する機能を有し、前記ファイル記憶制御手段は前記ファイル記憶手段が扱うデータビット数を前記中央演算処理装置の処理データビット幅に合わせるデータビット幅制御手段を有し、該データビット幅制御手段はデータビット幅を一致させるために必要とする期間、前記状態提示信号により処理中であることを前記中央演算処理手段に示して待機させ、双方の処理データ幅を一致させてデータのやり取りを行うことを特徴とした請求項5記載の情報処理装置。

【請求項7】前記中央演算処理手段のデータアクセス幅に等しいビット数のデータを生成するために必要な数のフラッシュメモリ群により前記ファイル記憶手段を構成し、それぞれのメモリ群にシーケンシャルな順序を規定し、複数の記憶容量単位数に及ぶファイルの格納においてはこの規定されたメモリ群の順序に従うこととし、一旦格納されたファイルを更新することにより記憶容量単位数が増加する場合には該増加に対して、前回格納したときにファイルの最後のデータが格納されていたメモリ群の次の順序に規定されているメモリ群から増加記憶容量単位分の記憶エリアを確保することを特徴とする請求項6記載の情報処理装置。

【請求項8】中央処理装置と、ファイル記憶装置を備える情報処理装置において、

前記ファイル記憶装置の記憶媒体として、複数のフラッシュメモリ素子を用い、該フラッシュメモリのデータの最小消去単位をファイル管理の記憶容量単位と等しくし、かつ、

前記中央処理装置がファイルアクセスを要求する際の記憶容量単位数が複数である場合は前記複数のフラッシュメモリ素子を同時にアクセスし、記憶容量単位数が1である場合には、前記複数のフラッシュメモリ素子の1個をアクセスすることを特徴とした情報処理装置。

【請求項9】フラッシュメモリを記憶媒体とするファイル記憶装置において、記憶するある1つのファイルの記憶容量が複数の記憶容量単位に及ぶ場合には、各記憶容量単位ごとに、次に続く記憶容量単位が格納されている物理的な位置の情報を記憶する連鎖情報記憶手段を備えたことを特徴とするフラッシュメモリファイル記憶装置。

【請求項10】1つのパッケージに複数のメモリチップが組み込まれているメモリ素子において、

前記メモリパッケージが全メモリチップのデータ入出力数の総計に相当する数の入出力データ端子を備え、また各メモリチップの入出力データとデータの入出力端子を切り換え接続制御するデータ制御手段と、該データ制御手段に対するメモリ使用者からの指示のための制御信号端子を備え、該制御信号端子の指示により前記複数のメ

メモリチップのデータを切り換え接続して前記入出力データ端子の任意の端子に任意のメモリチップの入出力データを接続することを可能としたメモリ素子。

【請求項11】1つのパッケージに複数のメモリチップが組み込まれているメモリ素子において、前記メモリパッケージが全メモリチップのデータ入出力数の総計に相当する数の入出力データ端子を備え、また各メモリチップの入出力データとデータの入出力端子を切り換え接続制御するデータ制御手段と、該データ制御手段に対するメモリ使用者からの指示のための制御コマンド設定手段を備え、該制御コマンド設定手段の指示により前記複数のメモリチップのデータを切り換え接続して前記入出力データ端子の任意の端子に任意のメモリチップの入出力データを接続することを可能としたメモリ素子。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、フラッシュメモリを記憶媒体とするファイル記憶装置を搭載した情報機器に係り、特に高速のファイルアクセスを要求する情報機器に適したファイル記憶装置のファイルアクセス方式に関する。

【0002】

【従来の技術】パーソナルコンピュータを始めとする今日の汎用の情報機器では補助記憶装置はほとんど必須といえる周辺機器である。しかも、補助記憶装置は、本体の機器に内蔵されているのが一般的であり、ユーザは常に大容量のファイルを扱うことができるようになっている。

【0003】最近では、ノートサイズやパームトップ型のパーソナルコンピュータが使われることが多くなってきており、移動して使用する用途が重要視され始めている。このため振動に弱く消費電力が大きい磁気ディスク記憶装置のかわりに半導体を記憶媒体としたファイル記憶装置が注目を浴びている。例えば、特開平2-292798号公報には、記憶媒体としてフラッシュメモリを使用した半導体ファイル記憶装置の技術が開示されている。

【0004】フラッシュメモリとは、電気的書換え可能な不揮発性メモリであり、大容量低価格化が可能であるため、半導体ファイル記憶装置の記憶媒体として最も有効なメモリの一つである。先記の特開平公報技術は、このメモリを使用してファイル記憶装置を構築する際に問題となる点を解決し、使い勝手を向上させる工夫をしている。例えば書換えの頻発により素子が劣化するというフラッシュメモリの欠点を救済する方法や、フラッシュメモリの書換えに必要な消去というフェーズにおける高速化の手法などを提案している。また、ホストとなる情報機器とのインタフェースとしては磁気ディスク装置と同一であることを提案しており、磁気ディスク装置に代

替できるシステムの構築を目的としている。

【0005】

【発明が解決しようとする課題】上記従来技術の半導体ファイル記憶装置はホストとなる情報機器の既存のインタフェースバスを使用し、磁気ディスク装置とのコンパチビリティを重視している。これにより半導体ファイル記憶装置をユーザに抵抗なく受け入れられるようにしているが、磁気ディスク装置との互換性を重視する反面、半導体記憶素子の磁気ディスク装置に対する優位性の利用に考慮がなされていない。

【0006】すなわち、半導体記憶素子は、回転するディスクからデータを拾い上げ書き込みを行う磁気ディスク装置と異なり、静的な記録媒体なので非常に高速なデータアクセスを可能とするが、磁気記憶装置と同一のインタフェースではこの高速アクセスの優位性を活かすことができないという問題がある。

【0007】本発明は、半導体記憶素子を記憶媒体とすることにより、アクセスの高速性能を最大限に引き出すインタフェースを構築し、磁気ディスク装置の欠点といえるアクセス性能の向上を新しい補助記憶装置を提供することにある。

【0008】

【課題を解決するための手段】本発明によるフラッシュメモリファイル記憶装置は、ファイルのデータをXビットのデータバスを介して格納するファイル記憶装置であって、最小消去単位が前記Xビットより大きくデータアクセス幅がxビット ( $x = X/p$  : pは2以上の整数) であるフラッシュメモリ素子を用いるフラッシュメモリファイル記憶装置において、それぞれ同時にアクセス可能なp組のフラッシュメモリ素子群により構成したフラッシュメモリ装置と、前記Xビットのデータバス上のデータを少なくともp分割する分割手段と、少なくとも、該分割により得られたp個のxビットデータを前記フラッシュメモリ素子群の1組に対応付ける第1の機能と、前記p個のxビットデータをそれぞれ前記フラッシュメモリ素子群の別個の組に対応付ける第2の機能とを有するデータ分配手段と、アクセス対象ファイルのファイル管理の記憶容量単位の数に応じて、前記第1および第2の機能を切替るよう前記データ分配手段を制御する制御手段とを備えることを特徴とする。

【0009】

【作用】現状のパーソナルコンピュータに使用されている磁気ディスク装置では、データアクセスがパーソナルコンピュータのメモリアccessに比較して遅く、パーソナルコンピュータ内のCPUと同期して動作する必要はない。そのため磁気ディスク装置のデータの授受は非同期のバス上で行われている。半導体を記憶媒体とした場合にはCPUの動作に追従することができるため、同期動作にすることが有意義となる。

【0010】しかしながらこの時の問題点として、CP

Uの処理のバス幅とフラッシュメモリ1チップのデータアクセスのバス幅が異なるという点がある。DRAMなどではこのバス幅の違いをメモリチップを並列して使用することにより解決できるが、フラッシュメモリでは消去単位が決まっており、最小の消去単位は例えば512バイトである。この場合、複数チップを並列に利用すると、一度に消去を行う消去単位の容量が(512×並列チップ数)バイトとなってしまう。これでは、消去対象容量が大きすぎて、消去対象以外のデータまで消去してしまうことになる。

【0011】一方、現在、パーソナルコンピュータのファイル管理はこの512バイト単位を一区切り(1セクタ)としているため、このファイル管理の記憶容量単位(以下、ファイル管理単位ともいう)を変更することは避けるべきである。従って、ファイル管理単位である1セクタのアクセスにおいて例えばフラッシュメモリ4チップを並列に利用しようとする、結果的に最小消去単位が大きくなり、このセクタの書替時に前述のように1セクタの4倍の容量を同時に消去することが必要になる。このような事態を避けるために、例えばCPUが32ビットバス、フラッシュメモリが8ビットバスであれば、1チップを4回順次アクセスしてCPUの32ビットバスに対応する必要がある。

【0012】しかし、4セクタ以上の連続アクセス時には、各4セクタのデータについて、4チップに並列に同時アクセスして、1回のアクセスでCPUのデータバスからの32ビットを同時アクセスするようにしても問題ない。すなわち、最初に第1セクタの512バイトが32ビットずつ16回に分けて書き込むと、各チップに128(16×8)バイトが分散する。次に第2セクタの512バイト32ビットずつ16回に分けて書き込む。続いて、同様に、第3セクタの512バイトを、さらに第4セクタの512バイトを書き込む。この時点で、4チップの最小消去単位の容量がすべてデータで満たされることになる。したがって、このファイルの書替時にはこの4セクタ分の容量を一括して消去して問題ない。

【0013】このようにして、ファイルを4セクタずつ同時4チップ並列アクセスを繰り返す。最後の4セクタに満たないデータが残った場合には、別の対処が必要となる。

【0014】3セクタ残った場合には、CPUの32ビットが3で割り切れないので、2セクタと1セクタに分けて対処する。

【0015】2セクタ残った場合、あるいは、初めから2セクタのファイルである場合、CPUからの32ビットのデータを2分割し、一度に16ビットずつ2チップに同時アクセスする。第1セクタの512バイトは、16ビットずつ32回に分けて2チップに書き込まれ、各チップに256(32×8)バイトずつに分散することになる。次に、第2セクタの512バイトが同様に16

ビットずつ32回に分けて2チップに書き込まれる。

【0016】1セクタ残った場合、あるいは初めから1セクタのファイルの場合、CPUからの32ビットのデータを4分割し、一度に8ビットずつ1チップにアクセスする。1セクタの512バイトは64回に分けてどのチップの1セクタエリアに格納される。

【0017】以上のように、連続アクセスするセクタ数によりデータの格納方法を変え、アクセスの高速化に寄与するとともにフラッシュメモリに伴う消去の問題を解決できる。すなわち、いずれの格納方法においても、1チップの最小消去エリア内には同一のファイルのデータのみが格納されることが保証されるので、フラッシュメモリの消去に伴う上述の問題が解消される。

【0018】なお、本発明では、フラッシュメモリを同期動作させるが、アクセス形式によって遅延が生じる。この問題に対しては、CPUにレディ信号入力があるものを用いることにより、任意にCPUに待機状態を要求できる。レディ信号入力のあるCPUは今日では一般的である。例えば現在の汎用のパーソナルコンピュータに搭載されているCPUとして最も一般的なINTEL社の16ビット処理以上のCPUには全て備えられている。このレディ信号をネグートすることによりCPUは処理サイクルを停止するため、アクセスデータの読み出し、書き込みが終了していなければこの信号をネグートするだけで良く、アクセスが終了したらアサートすればCPUは処理の実行を再開する。これらはファイル記憶装置がCPUと同期動作をして初めて可能となる制御であるため、この時に同期したクロックを双方に入力して同期動作させることは必須である。これらの制御により一回の処理におけるデータ幅がCPUとフラッシュメモリで異なる場合にも、データ幅が揃うまでCPUを待機させることにより容易に対応可能である。

【0019】ファイル管理上一度連続セクタで扱ったものを後から分割することは通常あり得ない。すなわち、ファイルシステムはファイル単位でしかアクセスされない。したがって、複数セクタの連続アクセスの場合は書き込みの時点で複数チップを同時アクセスして複数セクタを並列使用するようにすれば、読み出しの際に同様の並列アクセスをすれば書き込んだままのファイルのデータを得ることができる。

【0020】以上のように、本発明によればフラッシュメモリを記憶媒体とするファイル記憶装置を搭載した情報処理装置において、磁気ディスク装置と比較して優位性が大きいアクセス性能において、最大限の性能を引き出せるようシステムとの同期制御を行う。その際にシステムのCPUに入力するレディ信号により適宜CPUを待機状態においてタイミングを合わせる。システムのデータバス幅とフラッシュメモリ1チップのアクセスデータビット数が異なる場合でも、できるかぎり高速にアクセスすることが可能になる。

【0021】特にアクセス速度がユーザに認識されやすい大容量のファイルのリードライトすなわち連続セクタアクセスにおいて効果大きい。また、システムのデータバス幅とフラッシュメモリチップのアクセスデータビット数は用途、性能、時代によって異なるが、様々なものに対応できる柔軟性の高い構成を可能とする。またメモリをインタリーブアクセスして高速化する方法にも応用できる。

【0022】連鎖情報を備える構成によれば、システムからのアクセス手順を簡略化し高速アクセスを援助する。そしてファイル管理自体を簡略化でき、コントロール回路や制御プログラムなどの簡素化が図れる。

【0023】また、メモリ素子の中にデータ分配機能を備えることにより、周辺回路の削減、データ処理の高速化が図れる。

【0024】

【実施例】以下、本発明における実施例を図を用いて詳細に説明する。

【0025】まず、図1に標準的なパーソナルコンピュータの構成に本発明のフラッシュメモリファイル装置を設置したものを示す。図中、1はデータやプログラムの処理を司るCPUであり、そのデータバス幅は32ビットであるものとする。2はシステム全体の同期クロックを発生するクロック発生器である。3はシステム内部のバスでデータバス、アドレスバス、メモリコマンド、IOコマンドなどを含む。4は本発明のフラッシュメモリファイル装置のファイル管理やメモリ制御を行うファイル制御回路である。5は、フラッシュメモリファイル装置の記憶媒体となるフラッシュメモリアレイであり、1チップのアクセスデータビット数は8ビットであるとする。6はシステムの主記憶を管理制御する主記憶制御回路、7は主記憶であるDRAMである。8は周辺IOバスの制御回路であり、周辺IO装置の一つとして表示制御回路9と表示装置10が接続されている。この他の周辺IO装置として、通信装置11や外部大容量記憶装置12などが挙げられる。

【0026】通常、周辺IOバス制御回路8には別のクロック発生回路が内蔵されており、これら周辺IO装置はそのクロックの周期に基づいて動作する。しかし高速化のために内部システムバス3に直接接続して、CPU1と同期動作するものも考えられる。13は、内部システムバス3に接続された回路を同期させるためにCPU1を含む各回路に供給されるクロック信号である。但し、CPU1に与えられるクロックと全く同一である必要はなく、回路によっては分周したもので同期していれば良い。

【0027】14は、CPU1に入力されるレディ制御信号であり、各回路より出力された状態提示信号をレディ制御回路15で統括してCPU1に入力される。16は、本システムの利用者が所望の処理を指示するための

入力装置の制御回路であり、17は入力装置である。図では入力装置17をキーボードとし、その制御回路16はキーボードコントローラ(KBDC)としている。

【0028】次に、図1のシステムの動作を説明する。通常の動作時は、使用者から入力装置17により指示された処理を実行すべく主記憶7に格納されているプログラムやデータをCPU1が演算処理し、その結果を表示装置10に表示する。また必要であれば通信装置11を起動したり、大容量外部記憶装置12へ大容量データの格納を行う。そして、ファイルを引き出したり格納したりという動作の際にはファイル制御回路4とフラッシュメモリ5からなるフラッシュメモリファイル装置を動作させる。システムの立ち上げ時もここからシステムプログラムをロードすることになる。これらの動作の際には、クロック発生器2が発生するクロック13により同期動作し、ある回路がCPU1に待機を要求する必要性が生じたら、その回路がレディ制御回路15にCPU待機を要求し、レディ信号14のネゲートによりCPU1にそれが伝えられる。CPU1はレディ信号14が再度アサートされるまで待機を続けることになる。この時、ファイル制御回路4はCPU1に要求されるファイル数によりCPU待機時間を増減する制御を行えるようになっている。この仕組みを図2により説明する。

【0029】図2は、フラッシュメモリファイル装置の内部構成を説明する図である。図中、3、4、5、13は図1と同様のものであり、以下はファイル制御回路4内部の構成要素である。21は、システムバス3とインタフェースを行うレジスタ群であり、22はファイル制御回路4の状態をCPU1に報告するためのステータスレジスタ、23はアクセスするスタートセクタ番号の設定レジスタ、24はアクセスする最後のセクタ番号の設定レジスタ、25はCPU1が要求する処理をコマンドコードとして指示するコマンドレジスタ、26はシステムバスとのデータの授受を行う窓となるデータレジスタである。27は、ファイル制御回路4内部の制御を統括するコントローラであり、ワンチップマイコンなどのプログラム可能なインテリジェントなLSIが理想的である。28は記憶媒体であるフラッシュメモリ5の制御を行う制御回路、29はフラッシュメモリに書き込みあるいは読み出すデータの制御を行うデータ制御回路である。30はメモリのアクセスを高速に行うためのDMA制御回路であり、システムのクロック信号13は本回路に入力される。31はレディ制御回路15に出力する状態提示信号、32はこのフラッシュメモリファイル装置内のローカルバス、33はフラッシュメモリをアクセスするための制御信号及びアドレスである。

【0030】次に図2のフラッシュメモリファイル装置の動作を説明する。CPU1はフラッシュメモリファイル装置へのアクセスの必要性が生じたら、システムバス3を通してアクセスを行う。それにはまずステータスレ

ジスタ22の内容を読み出してアクセス可能な状況を確認する。そして次にアクセスするセクタをスタートセクタレジスタ23とエンドセクタレジスタ24に設定する。そして要求するアクセスのコマンドコード（リードまたはライト）をコマンドレジスタ25に書き込む。そして再びステータスレジスタ22を読み出し、アクセス可能であればデータレジスタ26へのデータの書き込みあるいは読み出しを行う。

【0031】この際、コントローラ27はこれらインタフェースレジスタ21の管理を行い、CPU1からの要求に応えるようにする。つまりスタートセクタレジスタ23やエンドセクタレジスタ24、コマンドレジスタ25を読み取ってフラッシュメモリ5のアクセス内容を把握し、現在の状態を示すコードをステータスレジスタ22に書き込んでCPU1に報告する。

【0032】コントローラ27はCPU1からのアクセス要求に応えるためのフラッシュメモリアccessを直接行うには動作速度が遅いことが考えられるので、DMA制御回路30により高速にフラッシュメモリアccessを行い、システムバス3とのデータの授受を行う構成としている。そのためのDMA制御回路30やメモリ制御回路28へのアクセス内容の設定やDMAの起動などをコントローラ27が行うことになる。

【0033】DMA制御回路30はDMAを行うためのアドレス発生及びタイミング生成を行い、メモリ制御回路28はそのタイミングに従いアクセス信号を発生する。フラッシュメモリ5はこれらの入力信号によりデータ制御回路29とデータの授受を行う。データ制御回路29はアクセスのセクタ数に応じてデータの生成を行う。

【0034】例えば、1セクタのライトアクセスであればシステムバス3からの1回のアクセスにより送られるデータをフラッシュメモリの1チップの書き込みビット数に合致させる。本実施例では、システムバス3のデータバス幅を32ビットとしているため1回のアクセスによる転送で32ビットのデータが得られる。そしてフラッシュメモリ1チップのデータ幅を8ビットとしているので送られたデータを4回に分けてフラッシュメモリ5に書き込むことになる。そのために32ビットデータを4つの8ビットデータへ分割する処理を、データ制御回路29がラッチ回路を用いて行う。また逆に1セクタのリードアクセスであれば、フラッシュメモリ5を4回リードアクセスして32ビットデータを用意して、1回のバス転送とする。この際に生ずるシステムバス3の待ち時間はデータ制御回路29が発生する状態提示信号31によるCPU1への待機要求で生じさせる。

【0035】一方、複数セクタのアクセスの場合は、データ制御回路29がラッチするデータを調節してアクセスを高速化して正常実行する。例えば4セクタの連続リードアクセスであれば、8ビットアクセスのフラッシュ

メモリ5を4チップ同時にリードすることにより32ビットのデータが完成するため、システムバス3の待ち時間を大きく減らすことができる。ただしフラッシュメモリ5への書き込みの時点で4チップ同時ライトアクセスを行っておく必要がある。そうしないとデータの順番が異なってしまい正常なファイルデータではなくなる。ただし、ファイル記憶装置を扱うシステムでは、ファイル単位で管理を行うのが一般的であるため、書き込んだときのセクタ数と同じセクタ数をリードアクセスするのが普通であり、セクタ数によるアクセスの形式をライトとリードで全く等しくしていれば先述の情報を記録する必要はない。つまり、例えば5セクタの連続ライトの際には最初の4セクタは4チップに並列に同時書き込みし、残る1セクタは1チップに4分割する、というアクセス形式をリードとライトのいずれにおいても採用すれば常に正常なファイルデータをアクセスできる。なお、念のために、フラッシュメモリ5に格納されているセクタ単位のデータに、データ格納形式の情報を記録しておいてもよい。その記録場所としてはフラッシュメモリのデータ格納領域以外の冗長領域が存在すればそこに格納するのが適当であり、冗長領域がない場合は別の記憶領域を設置して記録する。

【0036】6セクタの連続アクセスに際しては、4セクタの並列アクセスと2セクタの並列アクセスにより処理を行う。

【0037】次に、これらのアクセス信号およびデータ制御方法を図3によりさらに詳しく説明する。

【0038】図3では、システムバスを32ビット、フラッシュメモリを8ビットのビット幅としたときの構成を示している。図中、既出の番号はこれまでに説明したものと同一のものである。新たに、41は、DMA制御回路30のアドレス発生のためのカウンタであり、システムクロック13やシステムバス3のIOアクセス（コマンド）信号あるいはメモリアccess（コマンド）信号を入力し、これに同期してカウントアップする。42は、コントローラ27のローカルバス32が接続されるDMA制御の起動レジスタであり、このレジスタにコードを書き込むことにより所望のDMA転送を開始することができる。43は、やはりローカルバス32に接続されるセクタ番号レジスタであり、アクセスするセクタ番号を書き込むことにより任意のセクタ番号のDMA転送を行うことができる。実動作上は、このセクタ番号の書き込み値をメモリ制御回路28に入力してフラッシュメモリの上位アドレスやチップセレクト信号の生成に使用される。44は、DMA転送時に各制御回路で同期をとるためのタイミング信号を発生するタイミング回路である。45は、カウンタ41とセクタレジスタ43の値に基づいてメモリ制御回路28が発生するメモリアドレスである。46は、メモリアドレス45の発生に合わせて生成するメモリ制御信号である。51、52、53、5

4は、各々1バイト（8ビット）のデータラッチであり、データ制御回路29内における32ビットデータと8ビットデータとの間のデータ幅変換のための4バイト（32ビット）のデータラッチを構成している。システムバス3のデータをD0～D31とすると、ラッチ51はD0～D7、ラッチ52はD8～D15、ラッチ53はD16～D23、ラッチ54はD24～D31をそれぞれラッチする。55は、これらデータラッチのラッチ信号発生回路である。56は、ローカルバス32と接続されてデータ幅やデータの並び方を設定するデータ制御設定レジスタである。本実施例では、このデータ制御設定レジスタ56にリードアクセス時の連続アクセスセクタ数として“1”、“2”、“4”のいずれかを設定することによりラッチ信号の発生とそのタイミングをラッチ信号発生回路55に指示する。そしてデータラッチ51、52、53、54のそれぞれに入力するラッチ信号が57、58、59、60である。例えば“1”を設定すれば、ラッチ信号57、58、59、60を順次一つずつ出力し、1チップのフラッシュメモリからのデータを4回アクセスして32ビットのデータを生成してシステムバス3に出力する。また“2”を設定すれば、ラッチ信号57、58を同時に出力し、その後ラッチ信号59、60に同時に出力するということを交互に行い、2チップのフラッシュメモリからのデータを2回ずつアクセスして32ビットのデータを生成してシステムバス3に出力する。また“4”を設定すればラッチ信号57、58、59、60に同時に出力して、4チップのフラッシュメモリからのデータを1回だけアクセスして32ビットのデータを生成してシステムバス3に出力する。

【0039】一方、ライトアクセスの時は常にデータラッチ51、52、53、54に同時にラッチ信号を送ってシステムからの32ビットデータを一度にラッチすることになる。61は、データラッチ51、52、53、54に格納されたデータあるいはメモリ5からのデータを振り分けるデータ分配回路である。62は、フラッシュメモリ5とデータ分配回路61を結ぶ32ビットのデータバスである。63は、データ分配回路61のデータの方向を決定するためのリードライト信号であり、インタフェースレジスタ21のコマンドレジスタ25などから供給を受けるのが良い。データ分配回路61は双方向のパッファになっており、片側はデータラッチ51、52、53、54に接続され、もう一方はフラッシュメモリ5に接続されている。フラッシュメモリ5側では32ビット分の入力となっており、フラッシュメモリ5の全チップを4組に分けて1組ずつ別のビット群（ビット0～7）（ビット8～15）（ビット16～23）（ビット24～31）に分け、計32ビットとして入力している。そしてリードかライトかによって方向を定め、それぞれのデータの振り分けをデータ制御設定レジスタ56の設定内容によって決定する。

【0040】このデータの分配について図12、図13、図14により具体的に説明する。これらの図は、データ分配回路61のデータ分配例をデータ制御設定レジスタ56の設定値別に示したものである。それぞれ、図12は連続アクセスセクタ数が1、図13は連続アクセスセクタ数が2、図14は連続アクセスセクタ数が4、である場合におけるデータ分配を示しており、各場合に4つのメモリ群のどれにアクセスするかにより4種類のデータ分配がある。またリードライトも区別されている。連続アクセスセクタ数が1の場合には4つのシステムサイクルで1アクセスとなり、連続アクセスセクタ数が2の場合には2つのシステムサイクルで1アクセス、連続アクセスセクタ数が4の場合には1つのシステムサイクルで1アクセスとなる。

【0041】連続アクセスセクタ数が1の場合は4つに分割したメモリのどこにアクセスするかにより4種類に分けられるが、連続アクセス数が2及び4においては、どのメモリ群を起点にするかによりやはり4種類に分かれる。すなわちメモリの使われ方により起点を適宜決めることができ、このようにすることにより使用されるメモリ群が偏ることを防ぐことができる。例えば必ずメモリ群1を起点にするとデータの分配法は簡素化できるが、メモリ群1が使用される比率が高くなり使用量、使用頻度ともに偏りが生じることが予想できる。偏りが生じると最終的には、一つのメモリ群が使用できなくなってしまうことになる。従って起点は全てのメモリ群に設定可能とする。

【0042】なお、連続セクタ数が1または2のリードにおいては、図では分配回路のデータ結線をそれぞれのサイクルにおいて別々にしているが、ラッチ信号を目的のラッチ以外には出力しないため、各サイクルで分けることなく結線を一緒にしてもよい。つまり連続セクタ数が1（図12）のAのリードにおいては、分配を順次のサイクルで、ラッチ1—メモリ群1、ラッチ2—メモリ群1、ラッチ3—メモリ群1、ラッチ4—メモリ群1としているが、全てのサイクルで全てのラッチへの結線をメモリ群の1に接続しても構わない。これはデータの結線をしていてもラッチ信号を出力しなければ影響がないからである。

【0043】さらに具体的な説明を加えれば、例えば1セクタのライトアクセスでは図12を参照し、32ビットのデータバス62のうちアクセス対象のフラッシュメモリ群の1つに接続された1バイトを、1回目ではデータラッチ51に、2回目ではデータラッチ52に、3回目ではデータラッチ53に、4回目ではデータラッチ54に、以下51から順番に繰り返してフラッシュメモリへの書き込みを行う。リードにおいては、ラッチ信号により1サイクルずつ各ラッチにデータを振り分ける。

【0044】2セクタのライトアクセスでは、すなわち

データ幅2バイトのライトであれば、図13より32ビットのデータバス62のうち該当する2つのメモリ群に接続されている2バイトのデータとして1回目のアクセスではデータラッチ51、52より、2回目のアクセスでは53、54よりデータを受け取り、以下交互に振り分ける。一方、リードアクセスでは方向が逆になり、該当する2つのメモリ群から1回目のアクセスではデータラッチ51、52に、2回目のアクセスでは53、54にデータを振り分け、これを繰り返す。

【0045】4セクタのライトアクセス、すなわちデータ幅4バイトのライトであればバッファの方向はデータラッチからフラッシュメモリ5の方向となり、32ビットのデータバス62は先頭に当たるメモリ群から順にデータラッチ51、52、53、54に接続され、1サイクルのアクセスで32ビットのアクセスが完了する。リードでは方向が逆になる。

【0046】以上の動作を行うためにコントローラ27はDMA制御回路30に起動をかける前に、これまでに説明したレジスタ類に適切な値を設定しておく必要がある。

【0047】高速化のために、前述のように、複数のメモリ群に同時にアクセスを行う場合、各メモリ群に与えるアドレスについて、以下、図5を参照して検討する。

【0048】図5において、81～84は、それぞれメモリ群1～4を示している。同図(1)は、ある程度のファイルの書き込みが行われた状態を示し、同図(2)は、その後、あるファイルについて更新されそのファイル容量が増えた状態を示している。図示のデータ記号

(m-n)のmはファイル番号、nは各ファイル内のセクタ番号である。例えば、(3-2)は、ファイル番号3番の2セクタめを示す。但し、この図は、各ファイルの1以上のセクタに対して、どのメモリ群のエリア(1セクタ分の容量を有する)がどのように利用されるかを総体的に示したものであり、実際には、2以上のセクタの連続アクセス時には、各メモリ群の1セクタ容量エリア内に単独のセクタ内容のみが格納されるのではなく、複数のセクタの内容が分散して格納される。この点に関しては、ファイル管理の説明において後に詳述する。

【0049】図5(1)において、ファイル番号1(1-nと付されている全てのセクタ)のアクセスにおいて、1-1から1-4までの同時アクセスにおいては、これらが同じ列に並んでいるため、各メモリ群に同一のアドレスを与えることができる。しかし、ファイル番号2においては、2-1から2-4までを同時アクセス使用すると、メモリ群4には他のメモリ群とは異なるアドレスを与える必要がある。これは、アドレスバスをメモリの数だけ設ける必要があることを意味する。あるいは、メモリ群に異なるアドレスを与えるなんらかの方法が必要となる。従って、これを避けるためには、ファイル番号2のアクセスにおいては、2-1は単独でアクセ

スし、2-2、2-3を同時アクセスし、さらに2-4は再度単独でアクセスするというアクセス形式をとる必要がある。しかし、これでは高速化においては無駄が生じる。そこで、異なるアドレスを与える構成例を図9により説明する。

【0050】図9は、メモリ群数を4としたときのメモリ制御回路28内のアドレス生成回路の構成例である。図中、201～204はフラッシュメモリ5を構成する4つのメモリ群であり、201は第1のメモリ群a、202は第2のメモリ群b、203は第3のメモリ群c、204は第4のメモリ群dである。205は、メモリ群bに与える上位アドレスのラッチ回路b、206はメモリ群cに与えるラッチ回路c、207はメモリ群dに与えるラッチ回路dである。208は、メモリ制御回路28のアドレスバス(図3の45に対応)、209はアドレスバス208の下位アドレスである。そのアドレスビット数は、フラッシュメモリチップのデータ消去最小単位あるいはファイル管理の最小単位の記憶容量をアクセスするのに相当するアドレスビット数とする。例えば、データ消去最小単位が512バイトであれば、9ビットとなる。フラッシュメモリチップのデータ消去最小単位がファイル管理の最小単位よりも小さい場合に限り、ファイル管理の最小単位に相当するアドレスビット数とすることが有効である。210は、アドレスバス208の下位アドレス209を除いた上位アドレスである。但し、メモリ群をアクセスするのに不要なさらに上位のアドレスは除くものとする。211は、メモリ群bをアクセスするためにアドレスラッチbに記憶された上位アドレス、212はメモリ群cをアクセスするためにアドレスラッチcに記憶された上位アドレス、213はメモリ群dをアクセスするためにアドレスラッチdに記憶された上位アドレスである。214はメモリ群a 201のメモリ制御信号、215はメモリ群b 202のメモリ制御信号、216はメモリ群c 203のメモリ制御信号、217はメモリ群d 204のメモリ制御信号である。

【0051】図9の構成において、メモリ群aからメモリ群dまでを同時にアクセスする場合には予めアドレスラッチ205～207にそれぞれのメモリ群をアクセスする上位アドレスを書き込んでおく。そして、アクセスを行う際には、下位アドレス210はすべてのメモリ群に共通であるため、アドレスバスはメモリ群aだけをアクセスするためのアドレスを供給し、メモリ群a以外のメモリ群は対応するアドレスラッチ回路から上位アドレスによりアクセスを行う。なお、アクセスの制御はメモリ制御信号214～217に委ねられ、例えば、メモリ群aとメモリ群bだけのアクセスであれば、メモリ制御信号214、215だけがアクティブになるようにする。また、例えばメモリ群dだけのアクセスにおいては、メモリ群aにはメモリ制御信号214をアクティブにしなければいかなるアドレスが与えられても問題な



い。以上により、各メモリ群に異なるアドレスを与えることが可能となり、メモリ群さえ4つに分散していれば、物理的にアドレスの異なる箇所に格納されている同一ファイルのデータを同時にアクセスすることができ、高速化の貢献度を高めることができる。図9の構成では、メモリ群aのアドレスラッチを省略することができる。勿論、メモリ群aにもアドレスラッチを設けることが有効であれば、そのようにしてもよい。

【0052】以上の実施例によれば、CPU1は比較的小さいレジスタに簡単な指定を行うことで、所望のセクタのリードライトアクセスを高速にアクセスできる。また、ファイルコントローラをワンチップマイコンとしてきめ細かい制御をソフトウェアにより指示することができ、DMA転送制御回路を搭載することによりワンチップマイコンがシステムより遅い動作しかできなくてもデータ転送を高速に行える。ただし、もしワンチップマイコンの動作速度がシステムに対して十分に高速な動作が可能であるならば、DMA制御回路は必要なく全てのデータ転送をワンチップマイコンが行う構成も考えられる。

【0053】また、本実施例ではデータバスをシステム32ビット、フラッシュメモリ8ビットとして説明を行ってきたが、16ビット動作のCPU1や64ビット動作のCPU、また16ビット入出力のフラッシュメモリなどに対しても、データラッチの数やデータ制御設定レジスタなどの構成やコントローラの制御プログラムなどを変えることにより容易に他のデータ幅に対応できる。その具体例を図15に示した。図15は各ビット幅の組合せにおける、ハード構成の具体的数値を示している。図の横の構成はフラッシュメモリのデータのビット数であり、4ビット、8ビット、16ビット、32ビットを例に挙げている。縦の構成はシステムのデータ幅を示しており、簡単な情報機器の8ビットから、現在のパソコンの主流である16ビット、32ビット、そして将来的なパソコンや高性能コンピュータの64ビット、128ビットの例を示している。またハード構成として挙げているデータ分配回路は、実施例のデータ分配回路61で説明すれば、図中の円一つのビット数×円の数で示している。各円の接続は図12～図14と同様の考え方で示すことができる。ラッチ数は実施例におけるラッチ51～54の数の増減であるが、システムデータ幅よりメモリデータビット数の方が大きい場合にはラッチの位置はデータ分配回路メモリ側に構成されることになる。メモリ群数は実施例では4つに分けているメモリの分割数を示している。

【0054】ただし、本実施例を含めデータ分配回路の分配数やラッチ数、メモリ群数は最低限度の数であり、回路規模、端子数などに余裕があれば、増やすことによりさらに効果的なシステム構成となる。というのはメモリ群数を増やすことによりデータの格納メモリの選択に

余裕ができるため、先に述べたようなメモリの使用量、使用頻度の偏りをさらに防ぐことができる。また並列アクセスするメモリ数を増やすことができるため、高速化をさらに進めることができる。特にシステムデータ幅とメモリデータビット数が等しいかあるいはメモリのデータビット数の方が多い場合には、実施例で述べてきたようなデータバス幅を合わせるという効果は全くないが、複数のメモリチップを同時アクセスしていわゆるインタリーブ方式による高速化を実現することができる。この際にはメモリ群数を増やすことにより同時にデータ分配回路におけるメモリのデータの分配数や、ラッチ数を同時に増やす必要がある。

【0055】なお、本実施例ではフラッシュメモリがシステムからのデータ転送に追従できることとして説明しているが、そのためには、フラッシュメモリ内にライトバッファを搭載することが望ましい。今後、高速に書き込みを行えるフラッシュメモリが実現すれば、このようなライトバッファは必要なくなる。フラッシュメモリにライトバッファが内蔵されていない場合には、ライトバッファをフラッシュメモリとデータ制御回路の間に設置することができる。そしてライトアクセスの際には直接フラッシュメモリに書き込まずライトバッファに書き込みを行ってシステムからのデータ転送終了後にライトバッファからフラッシュメモリへの書き込みを行うことになる。この場合の実施例の構成図を図4に示した。

【0056】図4中、既出の番号はこれまでに説明したものと同じのものである。新出のものとして71はアクセスがリードかライトかによってデータ分配回路61との接続を切り換えるデータセクタ、72はライトデータを一時的に格納する32ビット幅のライトバッファ、73はデータセクタ71を切り換えるためのリードライト信号であり、コマンドレジスタ25のデータの一部を入力すると良い。もしアクセスがライトであればデータ分配回路61をライトバッファに接続し、ライトデータをライトバッファに格納してシステムからのデータ転送後コントローラがフラッシュへの書き込みを行う。リードであればデータ分配回路61がフラッシュメモリに直接接続され前述の図3のリードアクセスと同様の動作が実行される。以上の構成にすることにより、ライトバッファを内蔵しておらず、書き込みの遅いフラッシュメモリを使用してもシステムから見れば高速なアクセスが可能となる。

【0057】次にファイル管理の実施例として図5を例にして説明する。これまではハード構成の実施例を説明してきたが、實際上、フラッシュメモリにどのように格納するかについて以下に述べる。基本的にこの操作はシステムのCPU及びシステムプログラムとフラッシュメモリファイルのコントローラが行うことになり、ソフトウェアによる操作が中心となる。図5による説明ではそのソフトウェア自体の内容ではなく、その操作の結果と

して、フラッシュメモリの各メモリ群の単位容量エリア（1セクタ分）が総体的にどのようにファイルのセクタに割り当てられるかを示す。ファイルの各セクタに対して、順次のメモリ群に単位容量エリアが確保される。4以上の連続セクタに対しては、4セクタ（例えば、図の1-1~1-4）に対して、順次メモリ群1~4の1単位容量エリアが割り付けられる。但し、実際にはこれらの各セクタのデータは4つの単位容量エリア内に分散して格納される。2セクタ（例えば図の1-5~1-6）に対しては、2つのメモリ群の単位容量エリアが割り当てられる。この場合も各セクタのデータは2つの単位容量エリア内に分散して格納される。ファイル1の最後のセクタ1-7には単独のメモリ群の単位容量エリアが割り当てられる。

【0058】ここで、2セクタおよび4セクタの分散格納の様子について、図10および図11によりさらに詳細に説明する。図10は4バイト同時書き込みの例を示し、図11は2バイト同時書き込みの例を示す。前述と同様、ホストのシステムバスが32ビット、1メモリ群のメモリアクセス幅が8ビットとする。

【0059】図10において、メモリ群a~dへの4セクタ同時書き込みの例を示す。図中、211は、ホストのシステムバスからのデータを示し、32ビット（4バイト）を単位として送られてくる。222~225は、8ビット（1バイト）単位のデータの通し番号を示しており、222は1バイト目のデータ、223は2バイト目のデータ、224は3バイト目、225は4バイト目のデータである。以下、連続して4セクタ分として2048バイト目までが送られてくる。実際には、32ビットバスであるため、1バイト目から4バイト目までが同時に受領され、以下も同様である。51~54は図3で示したデータ一時記憶のためのデータラッチである。

【0060】4バイトずつの同時書き込みであるため、32ビットデータつまり4バイトを同時にデータラッチ51~54に格納した後、各バイトがメモリ群a~dに分散して1バイトずつ書き込まれる。次の32ビットデータについて各バイトがメモリ群a~dの後続部分に書き込まれる。このようにして、1セクタ分の512バイト目までが書き込まれると、続いて第2のセクタについて同様に4バイトずつメモリ群a~dの後続部分に書き込まれる。図5では、便宜上、各セクタを特定のメモリ群に割り当てるように図示したが、以上の説明のように、実際には、各セクタのデータは特定のメモリ群のみに格納されるのではなく、メモリ群a~dに分散して書き込まれる。ただし、4セクタ全体のデータがメモリ群a~dの4セクタ分のエリアに格納されることは事実である。

【0061】図11においては、メモリ群aとメモリ群bへの2バイト同時書き込みの例を示す。

【0062】2バイトずつの同時書き込みであるため、3

2ビットデータつまり4バイトを同時にデータラッチ51~54に格納した後、これが2バイトずつに二分され、前半の2バイトがまずメモリ群aとメモリ群bに1バイトずつ書き込まれ、次に後半の2バイトがメモリ群aとメモリ群bに1バイトずつ書き込まれる。このようにして、1セクタ分の512バイト目までが書き込まれると、続いて第2のセクタの1バイト目から同様にしてメモリ群a、bの後続部分に書き込まれる。この場合も、第1セクタは特定のメモリ群のみに格納されるのではなく、メモリ群a、bに分散して書き込まれ、同様に、第2セクタも第1セクタに続いてメモリ群a、bに分散して書き込まれる。

【0063】単一セクタの書き込みについては図示しないが、この場合には、同時にデータラッチ51~54の各バイトデータが順次あるメモリ群に格納される。この場合のみ1セクタのデータは分散されずに特定のメモリ群内に格納される。

【0064】以上から分かるように、重要なことは、以上の動作の結果、フラッシュメモリ素子の最小消去単位（ここでは1セクタ容量に相当する512バイト）のエリア内には異なるファイルのデータが混在しないことである。これが保証されることにより、ファイル書替時に別のファイルのデータを消去することなく、かつ、メモリの記憶エリアを有効に利用しながら複数バイト並列同時書き込みにより高速のファイルデータ格納を達成することが可能になる。

【0065】前述のように、図5（1）はある程度のファイルの書き込みが行われた状態、図5（2）はその後あるファイルについては更新されかつファイル容量が増えてしまった状態を示している。図5（1）では格納されるファイル番号順、セクタ番号順に異なるメモリが割り当てられるように格納する。管理上無駄がないように、原則として、詰めて空きがないようにする。

【0066】図5（1）に示したように、メモリ群に空きが生じないようにファイルのセクタを、順次、連続的に詰めて格納するものとする、ファイル更新によりファイル容量が増えた場合には図5（2）に示すように物理的な格納位置としては連続的に格納することができない場合が生じる。このような場合であっても、メモリ群の順番としては連続的にするように格納する。つまり、ファイル番号4番のファイルを例にすると、（1）では4-1から4-5までの5セクタが格納されているが、（2）で2セクタ追加されると、4-1がメモリ群3から開始されているため、4-6、4-7の増加分はメモリ群2、3のエリアが確保されることになる。セクタ数が増加したファイルの書き込み時には、この時には4-5と4-6と2セクタ連続セクタとして扱われ、4-7は単独セクタとして扱われることになる。つまり追加セクタに対しても既格納セクタと合わせて連続セクタとしての扱いを受けることになる。これらのファイル管理の格

納と読み出しをスムーズに行うためには、ハードウェア構成としてシステムからのアクセスセクタの指定は開始セクタとセクタ数にし、物理的な格納位置の把握はファイルシステム内部において行われた方が適当であると考えられる。つまりシステムからのアクセスファイルの指定は高速化のため極力簡略化すべきであるが、物理的な位置が分散していると物理的な位置の指定が複雑化してしまう。従って図2に示したエンドセクタ24はこの意味からは取り去るべきである。そしてファイルの連鎖を示す情報をメモリのデータ以外の情報を格納する冗長領域があればそこに、なければ他の記憶手段に格納することにより、ファイルの開始番号が指定されればそれに続くセクタ番号の物理的な位置が全て連鎖して明らかになるため、連続アクセスが可能となる。

【0067】図6には連鎖情報の例を示している。図中、85はメモリに格納されているファイルデータであり、ファイル番号4、セクタ番号5のデータである。86はそれに続くファイル番号4、セクタ番号6のデータである。そして87はさらにそれに続くファイル番号4、セクタ番号7のデータである。ただし、これらのデータはセクタ番号を順番にしているが、実際の格納状態としては混じり合っている場合がある。つまり、連続セクタアクセスにより複数チップ同時アクセスすることによりデータをスクランブルして格納している場合がある。もっともこのときにもバイトごとのデータ格納の順番付けが必要であるため、セクタ番号は非常に重要である。88はファイルデータ85が格納されているメモリ上の物理的な位置を示す物理アドレスであり、左側の数字“3”はメモリ群番号、右側の数字“5”はそのメモリ群の中のアドレスである。89、90は同様にファイルデータ86、ファイルデータ87の物理アドレスである。91はファイルデータ85の次のセクタが格納されている物理アドレスを示す連鎖情報であり、ファイルデータ85がファイル番号4、セクタ番号5のデータであるため、連鎖情報91はファイル番号4、セクタ番号6のデータが格納されている物理アドレスを示していることになる。左側の数字がメモリ群番号、右側がメモリ群の中のアドレスである。同様にして、連鎖情報92はファイル番号4、セクタ番号7のデータが格納されている物理アドレスを示しており、連鎖情報93は次のセクタが存在しない場合を示している。つまりファイル番号4のファイルは7セクタで終了することがわかる。この連鎖情報がファイルデータに添付されていれば、システムのCPUはファイル格納の物理的な位置を把握する必要はなく単にファイル自体をアクセスする形式をとれる。そしてファイル制御のコントローラが連鎖情報を参照しながらメモリの物理的なアクセスを行うことにより、一つのファイルでありながら物理的な格納位置が連続しておらず分散しているファイルでも連続的なアクセスが可能となる。その際、異なるメモリ群の同時アクセスのために

異なるアドレスを入力する構成が必要である。ただし、図9で前述したように、1セクタアクセス分の下位アドレスは共用でき、それ以上の上位アドレスに関してだけで十分である。以上、連鎖情報を用いる実施例によればシステムからのアクセス指定を簡略化でき、少ない情報量で実施可能である。

【0068】次にメモリ素子自体に各種機能を組み込む実施例について説明する。

【0069】図7は複数のメモリチップを1つのパッケージに収め、それに付加回路を設けたメモリ素子の構成図である。図中、101はメモリパッケージ、102から105は同様の機能を持ったメモリチップ、106から109はメモリパッケージ101の入出力データ端子で、それぞれ一つで一つのメモリチップのデータ入出力が可能である。例えばメモリチップ102が8ビットのデータ入出力であれば106から109まで全部で8ビット×4=32ビットのデータ入出力端子を持つことになる。110はこれらのメモリチップと入出力端子を接続する経路を指定する信号入力端子、111は指定信号、112は接続経路設定回路である。本メモリ101のユーザは、データ入出力端子106から109の設定を信号入力端子110により行ってからアクセスを実行する。この設定には先述の図12、図13、図14のような接続経路から選択するものが一例として考えられる。メモリ101では設定されたデータ接続を指定信号111として接続経路設定回路112に送り、接続設定回路112では設定に従ってメモリチップ102から105をデータ入出力端子106から109に接続し、所望のデータバスにメモリチップのデータを接続することを実現する。

【0070】図8は、ユーザからのデータ接続の設定をコマンド入力により行うメモリ素子の構成例である。図中、113はコマンド設定値のレジスタとその設定値から指定信号111を作り出すコマンド制御回路である。114はデータ入出力端子の一部を使用してユーザがコマンド設定値を設定するデータ線である。その他は図7の同じ番号と同一のものである。ユーザはデータの接続経路を図12、図13、図14を一例として選択し、コマンドコードとしてコマンド制御回路113にデータバスの一部114から設定し、コマンド制御回路113では設定されたコマンドコードから接続経路設定回路112に該当する指定信号111を送る。以下は、図7の説明と同様である。図7、図8による本実施例によれば、メモリ素子としてこれまでに説明した他の実施例を実現することができ、周辺回路の削減を図ることができる。なお本実施例ではメモリパッケージ内に複数のチップと制御回路を搭載することとしているが、これらを一つにチップにまとめることにより小型化と高速化を助長できる。

【0071】

21

【発明の効果】本発明によれば、フラッシュメモリ装置において、ファイル書替時に別のファイルのデータを消去することなく、かつ、メモリの記憶エリアを有効に利用しながら複数バイト並列同時書込みにより高速のファイルデータ格納を達成することが可能になる。

【図面の簡単な説明】

【図1】本発明を実現するシステム構成を示すブロック図

【図2】フラッシュメモリファイル装置の実施例の構成を示すブロック図

【図3】実施例の動作を説明するための主要部分のブロック図

【図4】実施例における書き込みが遅いフラッシュメモリチップを使用した場合のデータ制御を示す説明図

【図5】実施例におけるメモリへのファイル格納の実行例の説明図

【図6】本発明の他の実施例における連鎖情報格納の実行例の説明図

【図7】信号端子入力指定によるデータ分配機能をメモリ素子内に備えたメモリ構成例の説明図

【図8】コマンド設定入力指定によるデータ分配機能をメモリ素子内に備えたメモリ構成例の説明図

【図9】実施例におけるメモリ群のアドレス接続の説明図

【図10】実施例における4バイト同時書込み時の動作の説明図

22

【図11】実施例における2バイト同時書込み時の動作の説明図システムバス32ビット、

【図12】実施例におけるメモリ1チップ同時アクセスのデータ分配の説明図

【図13】実施例におけるメモリ2チップ同時アクセスのデータ分配の説明図

【図14】実施例におけるメモリ4チップ同時アクセスのデータ分配の説明図

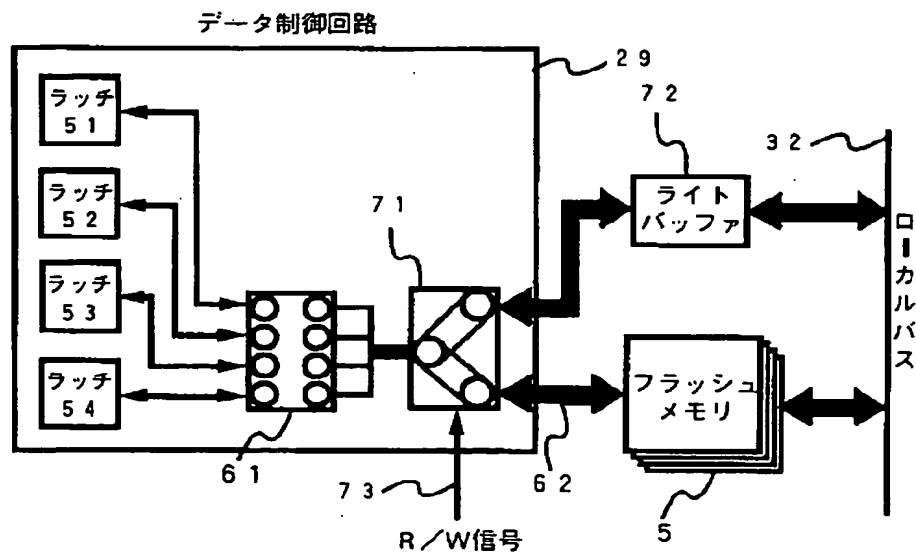
【図15】システムバスとメモリのデータバスの様々な構成におけるハード構成の説明図

【符号の説明】

2…クロック発生回路、4…ファイル制御回路、5…フラッシュメモリ、15…レディ制御回路、21…I/Fレジスタ、22…ステータスレジスタ、23…スタートセクタレジスタ、24…エンドセクタレジスタ、25…コマンドレジスタ、26…データレジスタ、27…コントローラ、28…メモリ制御回路、29…データ制御回路、30…DMA制御回路、31…状態提示信号、51…ラッチ回路、52…ラッチ回路、53…ラッチ回路、54…ラッチ回路、55…ラッチ信号発生回路、56…データ制御設定レジスタ、61…データ分配回路、71…データ切り換え回路、72…ライトバッファ、91…連鎖情報、101…メモリパッケージ、110…指定信号入力端子、112…接続経路設定回路、113…コマンド制御回路

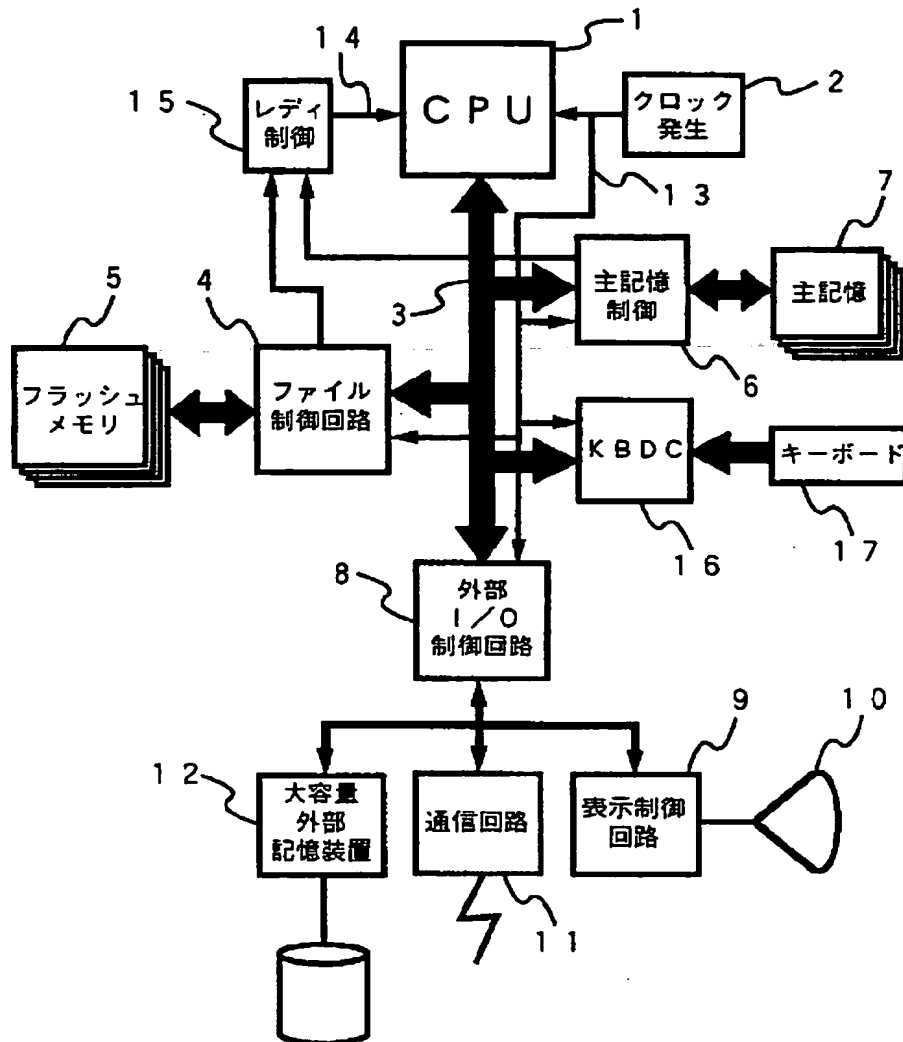
【図4】

図4



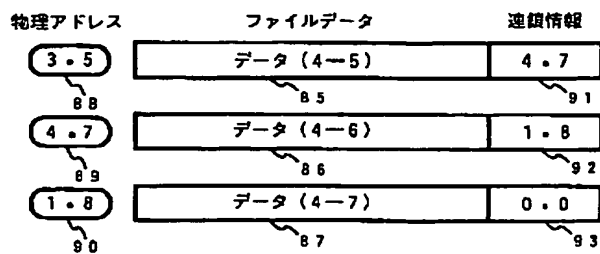
【図1】

図1



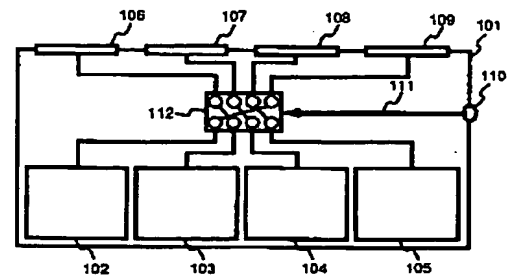
【図6】

図6



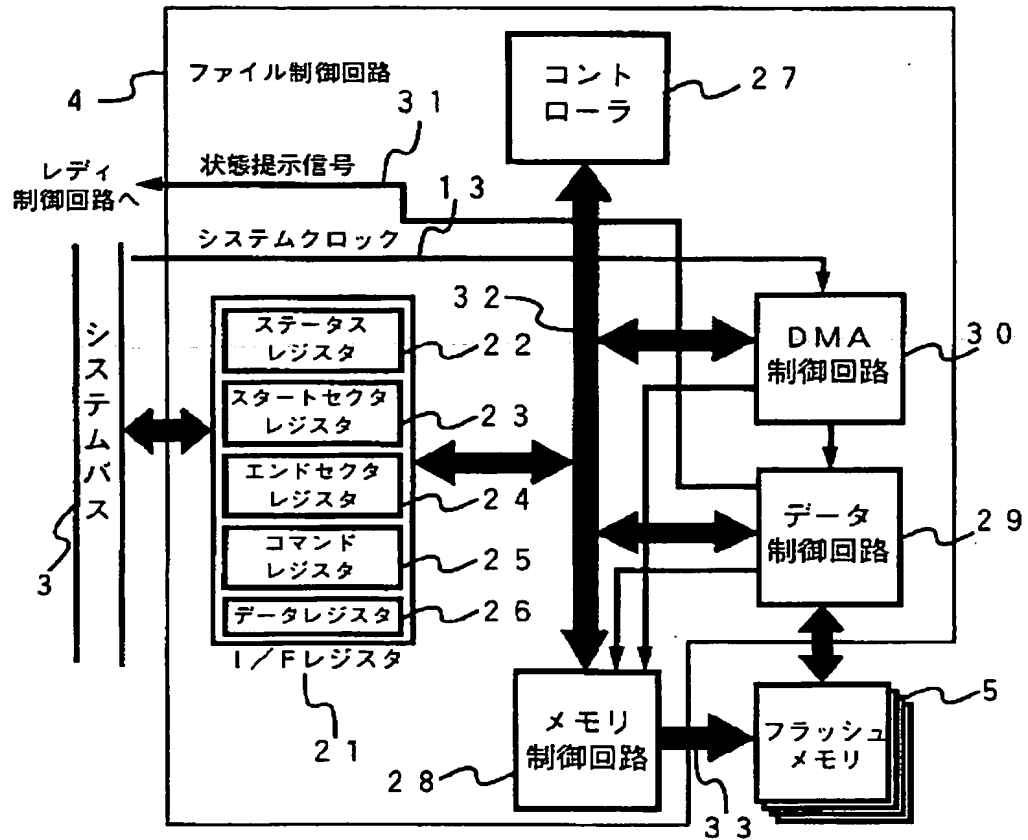
【図7】

図7



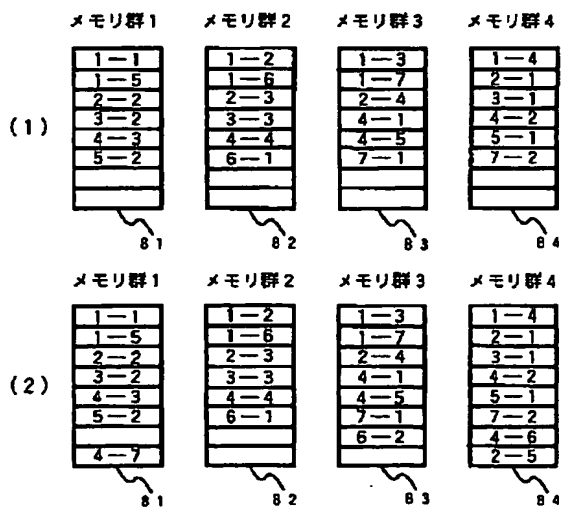
【図2】

図2



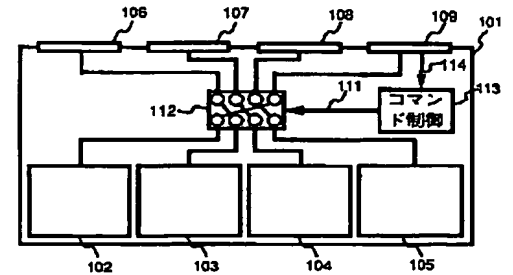
【図5】

図5



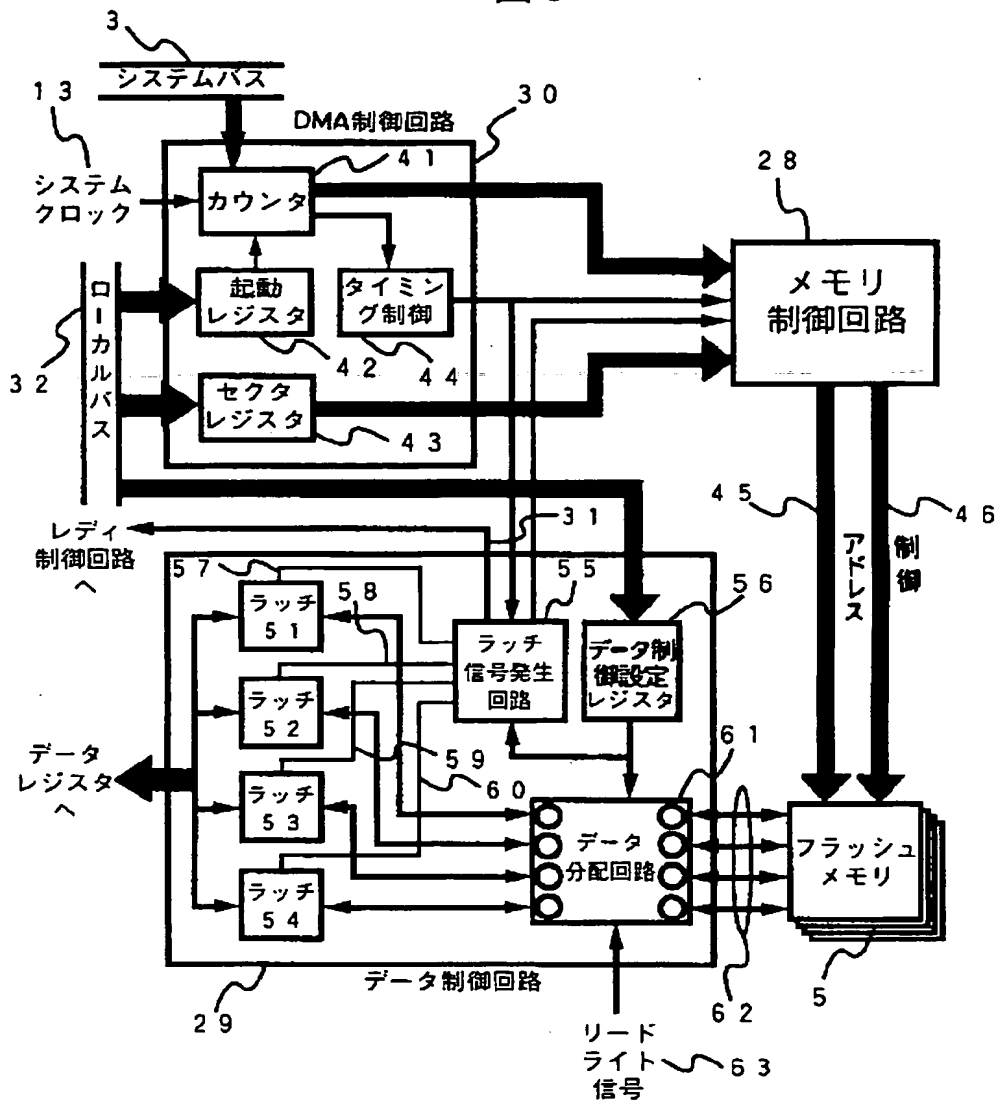
【図8】

図8



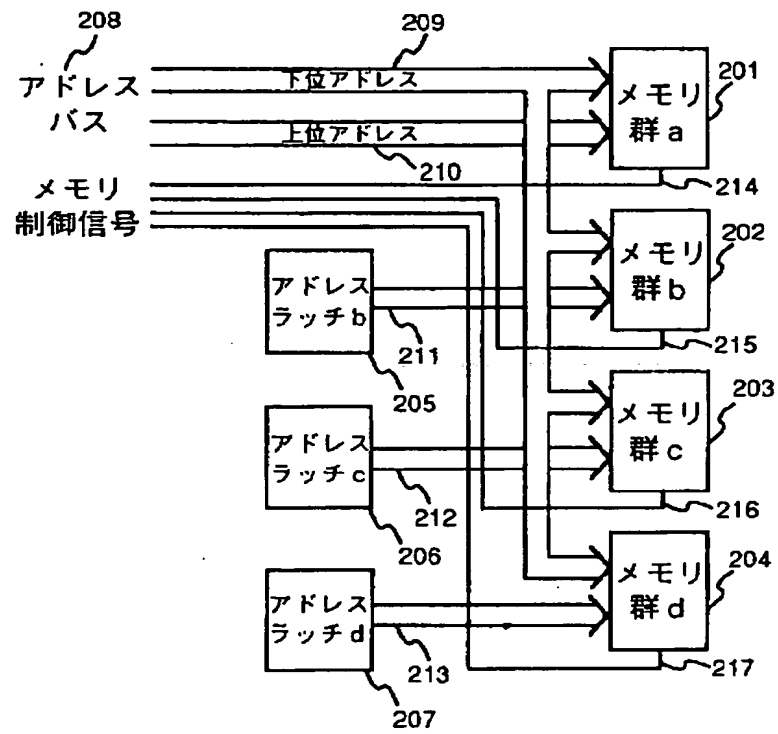
【図3】

図3



【図9】

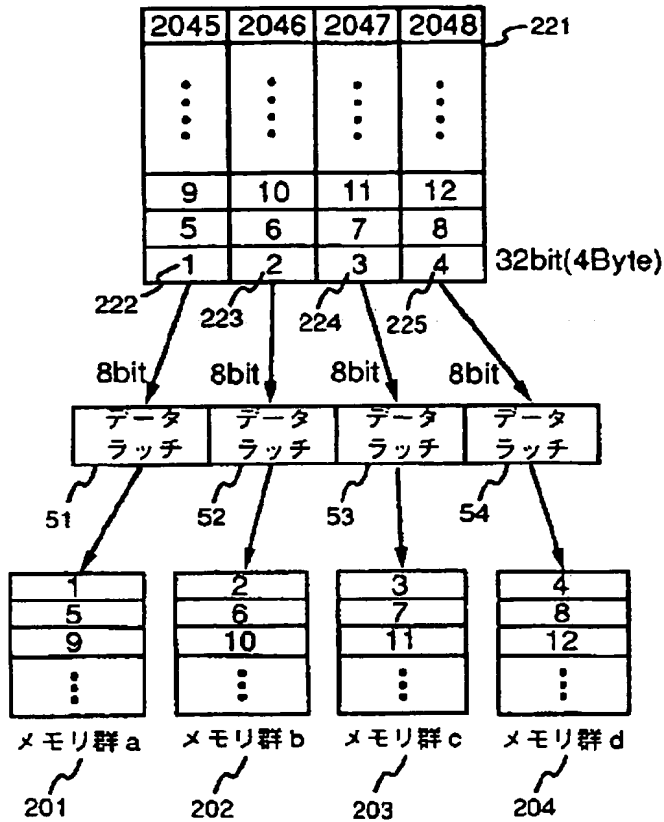
図 9





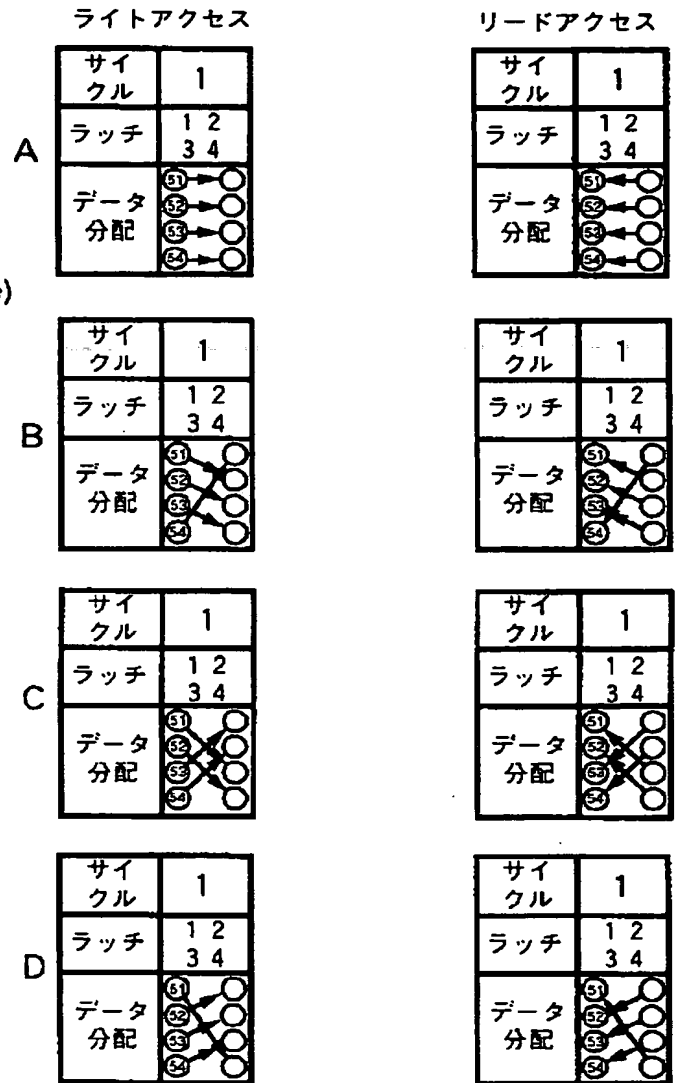
【図10】

図10



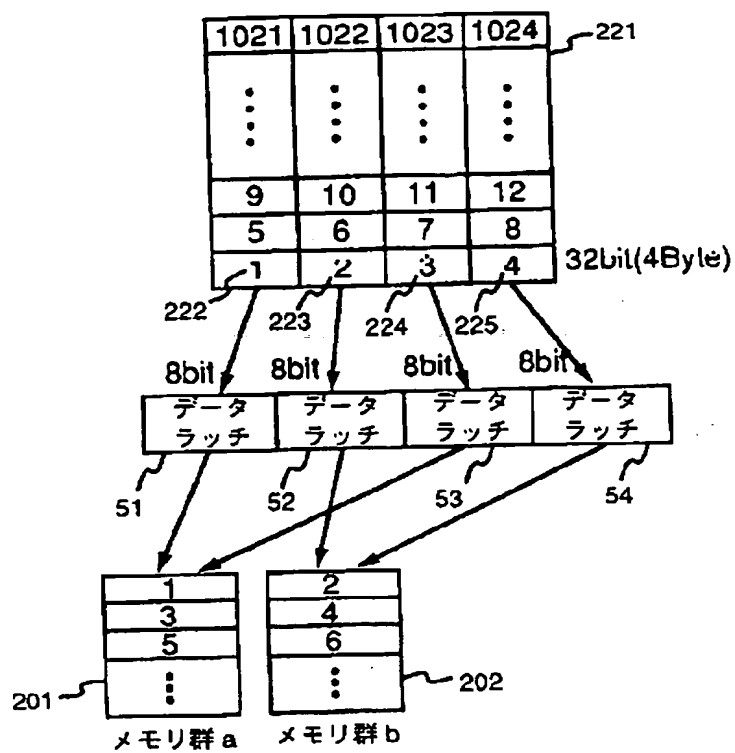
【図14】

図14



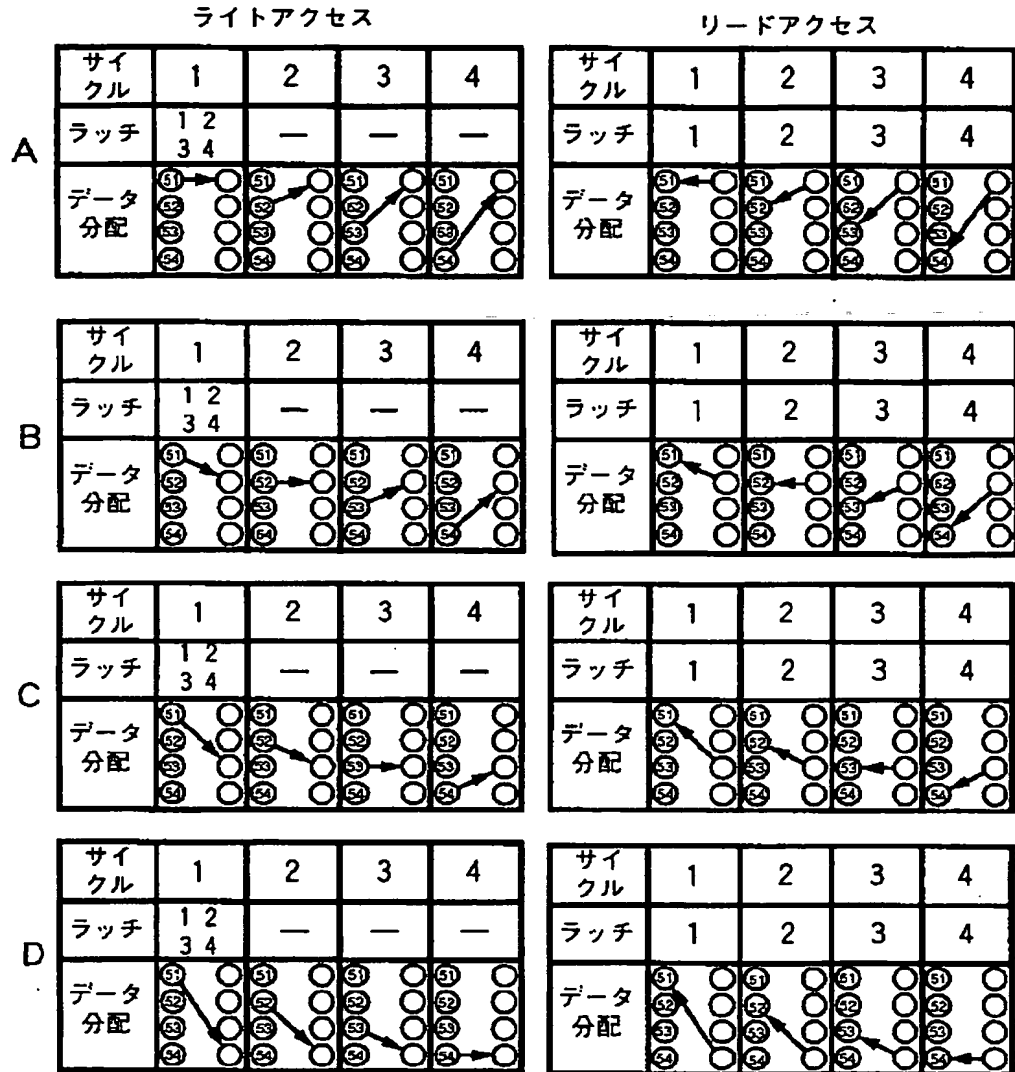
【図11】

図11



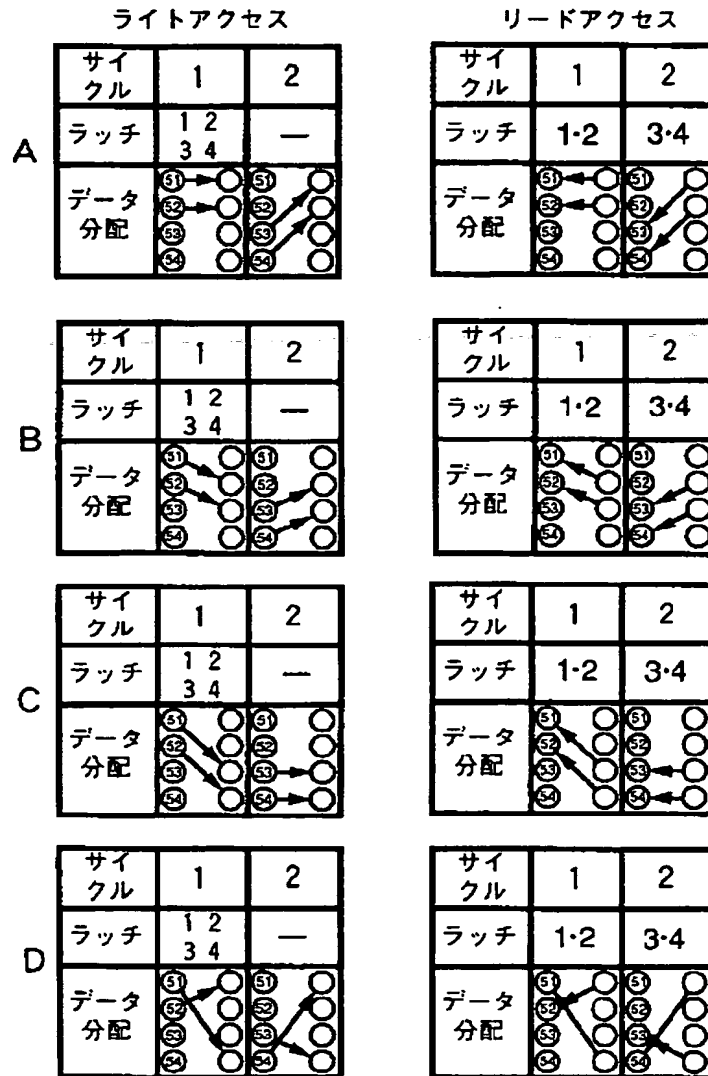
【図12】

図12



【図13】

図13



【図15】

図15

システム データ バス幅	メモリデータ ビット ハード構成	4 bit	8 bit	16 bit	32 bit
8 bit	データ分配回路	4 bit×2	8 bit×1	8 bit×2	8 bit×4
	ラッチ数	2	1	2	4
	メモリ群数	2	1	1	1
16 bit	データ分配回路	4 bit×4	8 bit×2	16bit×1	16bit×2
	ラッチ数	4	2	1	2
	メモリ群数	4	2	1	1
32 bit	データ分配回路	4 bit×8	8 bit×4	16bit×2	32bit×1
	ラッチ数	8	4	2	1
	メモリ群数	8	4	2	1
64 bit	データ分配回路	4 bit×16	8 bit×8	16bit×4	32bit×2
	ラッチ数	16	8	4	2
	メモリ群数	16	8	4	2
128 bit	データ分配回路	4 bit×32	8 bit×16	16bit×8	32bit×4
	ラッチ数	32	16	8	4
	メモリ群数	32	16	8	4

フロントページの続き

(72)発明者 北原 潤  
神奈川県横浜市戸塚区吉田町292番地 株  
式会社日立製作所マイクロエレクトロニク  
ス機器開発研究所内

(72)発明者 飛田 庸博  
神奈川県横浜市戸塚区吉田町292番地 株  
式会社日立製作所マイクロエレクトロニク  
ス機器開発研究所内

(72)発明者 古沢 和則  
東京都小平市上水本町五丁目20番1号 株  
式会社日立製作所半導体設計開発センタ内